frontiers | Frontiers in Communications and Networks

TYPE Original Research
PUBLISHED 17 June 2025
DOI 10.3389/frcmn.2025.1604850

OPEN ACCESS

EDITED BY
Ramoni Adeogun,
Aalborg University, Denmark

REVIEWED BY
Ahmed Aftan,
Middle Technical University, Iraq
Ahmad Bazzi,
New York University Abu Dhabi, United Arab Emirates

*CORRESPONDENCE
Zhaoting Liu,
liuzhaoting@163.com

RECEIVED 02 April 2025
ACCEPTED 19 May 2025
PUBLISHED 17 June 2025

CITATION
Qin Z and Liu Z (2025) Distributed quantile regression over sensor networks via the primal–dual hybrid gradient algorithm. *Front. Commun. Netw.* 6:1604850. doi: 10.3389/frcmn.2025.1604850

Distributed quantile regression over sensor networks via the primal–dual hybrid gradient algorithm

Zheng Qin and Zhaoting Liu*

School of Communication Engineering, Hangzhou Dianzi University, Hangzhou, China

As one of the important statistical methods, quantile regression (QR) extends traditional regression analysis. In QR, various quantiles of the response variable are modeled as linear functions of the predictors, allowing for a more flexible analysis of how the predictors affect different parts of the response variable distribution. QR offers several advantages over standard linear regression due to its focus on estimating conditional quantiles rather than the conditional mean of the response variable. This paper investigates QR over sensor networks, where each node has access to a local dataset and collaboratively estimates a global QR model. QR solves a non-smooth optimization problem characterized by a piecewise linear loss function, commonly known as the check function. We reformulate this non-smooth optimization problem as the task of finding a saddle point of a convex–concave objective and develop a distributed primal–dual hybrid gradient (dPDHG) algorithm for this purpose. Theoretical analyses guarantee the convergence of the proposed algorithm under mild assumptions, while experimental results show that the dPDHG algorithm converges significantly faster than subgradient-based schemes.

KEYWORDS

primal–dual, quantile regression, sensor networks, distribution estimation, robustness

1 Introduction

Distributed signal processing (Hovine and Bertrand, 2024; Cattivelli and Sayed, 2010; Schizas et al., 2009) in wireless sensor networks addresses the challenges of limited energy, processing power, and communication range of individual sensors. By applying collaborative computational algorithms, sensors can operate as a distributed signal processor, overcoming individual limitations and improving energy efficiency. Distributed signal processing is particularly used in various applications where distributed algorithms can enhance the power efficiency by avoiding data centralization, including environmental monitoring, healthcare, and military surveillance. However, the complexity of real-world sensor data often involves non-linear relationships, heteroscedasticity, and outliers. In such environments, traditional distributed methods like least squares regression may fail to provide robust estimates due to the influence of extreme values or noise.

Quantile regression (QR) (Waldmann, 2018) offers a solution by estimating conditional quantiles of the data distribution, rather than just the mean, making it more robust to outliers and better suited for modeling heterogeneous data sources. Quantile regression has gained significant attention as a robust approach to regression analysis, particularly

in situations where the distribution of the response variable is not symmetric or when outliers are present, and has applications in various fields, including ecology, economics, and industry (Cade and Noon, 2003; Ben Taieb et al., 2016; Wan et al., 2017). Some efficient numerical methods, including alternating direction method of multipliers (ADMM) (Mirzaeifard et al., 2024; Bazzi and Chafii, 2023), majorize–minimize (MM) (Kai et al., 2023; Cheng and Kuk, 2024), and machine learning (Patidar et al., 2023; Hüttel et al., 2022), were used for solving the optimization problem associated with quantile regression. Recent research has focused on distributed quantile regression (dQR) in sensor networks. In distributed sensor networks, where data from different sensors can vary significantly in terms of noise and variability, quantile regression can be applied at the local level to estimate the distributional characteristics of the data at each sensor node. Wang and Li (2018) proposed a diffusion-based distributed strategy [including a variant for sparse models (Bazzi et al., 2017)] for quantile regression over wireless sensor networks. Wang and Lian (2023), Lee et al. (2018), and Lee et al. (2020) introduced several consensus-based dQR methods for sensor networks. These methods overcome challenges in distributed settings, including limited storage and transmission power, while maintaining statistical robustness. They offer promising solutions for quantile-based analyses in decentralized sensor networks across diverse applications.

It should be noted that quantile regression involves a non-differentiable optimization problem with a piecewise linear loss function, also known as the check function. Most existing quantile regression algorithms rely on subgradient methods, which typically exhibit sublinear convergence rates. Although these methods have certain merits, they often struggle with slow convergence when tackling the non-differentiability of the optimization problem. Alternatively, techniques such as MM (Kai et al., 2023; Cheng and Kuk, 2024) mitigate the non-differentiability issue by minimizing a smooth majorizer of the check function instead of the function itself. However, these methods can introduce additional computational complexity and may not fully exploit the structure of distributed settings. As a data-driven approach, the machine learning-based methods (Patidar et al., 2023; Hüttel et al., 2022; Delamou et al., 2023; Njima et al., 2022) can circumvent the non-differentiability issue. However, it relies heavily on the availability of large datasets, which may not always be feasible or efficient in certain scenarios.

In this paper, we propose a novel approach for diffusion-based distributed quantile regression, leveraging the primal–dual hybrid gradient method to find a saddle point of a convex–concave objective. This strategy accelerates convergence and significantly enhances the efficiency of the quantile regression process.

2 Network model and problem formulation

2.1 Preliminaries

In this section, we present a brief introduction to quantile regression. Let S be a scalar random variable, B a L-dimensional random vector, and $F_S(s \mid \boldsymbol{b}) = P(S \le s \mid B = \boldsymbol{b})$ represent the conditional cumulative distribution function. The conditional quantile τ is defined as follows:

$$q_\tau^S(\boldsymbol{b}) = \inf\{s\colon F_S(s \mid \boldsymbol{b}) \ge \tau\}$$

for $\tau \in (0,1)$. A linear model is given by $s = \boldsymbol{b}^\top \boldsymbol{w} + \epsilon$, where $\boldsymbol{b}$ is the $L \times 1$ input data vector, $\boldsymbol{w}$ is the $L \times 1$ deterministic unknown parameter vector of interest, and ϵ is the observed noise following a certain distribution. Unlike standard regression methods, which focus on estimating the mean of $\boldsymbol{s}$, quantile regression provides a more comprehensive analysis by modeling different points (quantiles) in the distribution of $\boldsymbol{s}$.

In the quantile regression model, $q_\tau^S(\boldsymbol{b})$ is assumed to be linearly related to $\boldsymbol{b}$ as follows: $q_\tau^S(\boldsymbol{b}) = \boldsymbol{b}^\top \boldsymbol{w} + q_\tau^\epsilon$, where $q_\tau^\epsilon \in \mathbb{R}$ represents the τ-th quantile of the noise. The τ-th quantile of the noise, q_τ^ϵ, is not necessarily 0 (e.g., for asymmetric noise distributions). Explicitly including q_τ^ϵ avoids the restrictive assumption that $q_\tau^\epsilon = 0$, thereby allowing the model to flexibly adapt to the true noise distribution. Omitting q_τ^ϵ would force the conditional quantile $q_\tau^S(\boldsymbol{b})$ to pass through the origin, which is often inappropriate in real-world applications. Since both q_τ^ϵ and $\boldsymbol{w}$ are unknown parameters that must be estimated, the optimization problem for estimating $\boldsymbol{w}$ and q_τ^ϵ can be expressed as follows:

$$\{\boldsymbol{w}, q_\tau^\epsilon\} = \arg\min_{\boldsymbol{w}, q_\tau^\epsilon} \ \mathbb{E}[\rho_\tau(s - \boldsymbol{b}^\top \boldsymbol{w} - q_\tau^\epsilon)]. \tag{1}$$

Here, ρ_τ is the non-differentiable quantile loss function, also known as the check function, defined as follows:

$$\rho_\tau(u) = \begin{cases} \tau u, & \text{if } u \ge 0, \\ (\tau - 1)u, & \text{if } u < 0. \end{cases}$$

This function adjusts the loss asymmetrically depending on whether the residual u is positive or negative, allowing the model to estimate conditional quantiles for different τ values.

2.2 Network model and problem formulation

Consider a network consisting of K nodes distributed over a certain geographic region. Assume that the network is strongly connected, that is, there is no isolated node in the network. Every node $k \in \{1, 2, \ldots, K\}$ has access to the realization of zero-mean random data $\{s_{k,i}, \boldsymbol{b}_{k,i}\}$ at every time instant i and is allowed to communicate only with its neighbors $\mathcal{N}_k$, where $s_{k,i}$ is a scalar measurement, $\boldsymbol{b}_{k,i}$ is an $L \times 1$ measurement vector, and $\mathcal{N}_k$ denotes a set of nodes in the neighborhood of node k including itself. Moreover, $\{s_{k,i}, \boldsymbol{b}_{k,i},\ k = 1, \ldots, K,\ i = 1, \ldots, M\}$ satisfy a standard lineal regression model

$$s_{k,i} = \boldsymbol{b}_{k,i}^\top \boldsymbol{w}_0 + \epsilon_{k,i}, \tag{2}$$

where $\epsilon_{k,i}$ is the measurement noise and $\boldsymbol{w}_0$ is a deterministic sparse vector of dimension L.

The aim of this study is to develop a distributed quantile regression algorithm to estimate $\boldsymbol{w}_0$ using the dataset $\{s_{k,i}, \boldsymbol{b}_{k,i},\ k = 1, \ldots, K,\ i = 1, \ldots, M\}$. Recalling Equation 1, the sparsity-penalized quantile regression estimate of $\boldsymbol{w}_0$ is

obtained by minimizing the global cost function for quantile regression across the network, formulated as follows:

$$\min_{\bar{\boldsymbol{\omega}}} J^{glob}(\bar{\boldsymbol{\omega}}) \triangleq \min_{\bar{\boldsymbol{\omega}}} \frac{1}{K} \sum_{k=1}^{K} J_k^{loc}(\bar{\boldsymbol{\omega}}), \tag{3}$$

where

$$\begin{aligned} J^{loc}(\bar{\omega}) &\triangleq \frac{1}{M} \sum_{i=1}^{M} \sum_{l \in \mathcal{N}_k} c_{l,k} \rho_\tau \left(s_{i,l} - \boldsymbol{b}_{i,l}^\top \boldsymbol{w} - q_\tau^\epsilon \right) + \lambda \|\boldsymbol{w}\|_1 \\ &= \frac{1}{M} \sum_{i=1}^{M} \sum_{l \in \mathcal{N}_k} c_{l,k} \rho_\tau \left(s_{i,l} - \boldsymbol{a}_{i,l}^\top \bar{\omega} \right) + \lambda \|\boldsymbol{w}\|_1. \end{aligned} \tag{4}$$

Here, $J^{glob}(\bar{\omega})$ represents the global cost function over the network, and the local cost functions $J_k^{loc}(\bar{\omega})$, which reflect the costs at each node k, are aggregated to form the global objective. The first term in Equation 4 captures the quantile regression residuals across all nodes k and data points i, while the second term imposes ℓ_1-norm regularization on $\boldsymbol{w}$, encouraging sparsity. In this formulation, $\boldsymbol{a}_{k,i} = [\boldsymbol{b}_{k,i}^\top, 1]^\top$ is the augmented input data vector, $\bar{\boldsymbol{\omega}} = [\boldsymbol{w}^\top, q_\tau^\epsilon]^\top$ is the augmented parameter vector to be estimated, and $\boldsymbol{C}$ is a $K \times K$ weighting matrix with individual entries $\{c_{l,k}\}$. The coefficients $c_{l,k}$ $(l, k = 1, 2, \ldots, K)$ are non-negative weighting factors, satisfying

$$c_{l,k} = 0 \quad \text{if} \quad l \notin \mathcal{N}_k \quad \text{and} \quad \sum_{l=1}^{K} c_{l,k} = 1. \tag{5}$$

This condition (Equation 5) is explicitly satisfied by widely used weight rules in the distributed optimization literature (Cattivelli and Sayed, 2010; Tu and Sayed, 2011). In addition, λ is a regularization parameter controlling the sparsity of the solution, and $\|\boldsymbol{w}\|_1$ denotes the ℓ_1-norm which encourages sparse solutions.

We assume that the underlying network operates under ideal and stable communication conditions. Transient link or node failures—well-studied in the distributed network literature (Gao et al., 2022; Swain et al., 2018)—are effectively handled using established engineering solutions, such as fault-tolerant protocols, redundancy, and consensus mechanisms, thereby ensuring system reliability.

3 Distributed primal–dual hybrid gradient algorithm

This section first introduces a distributed quantile regression framework and formulates our problem as a saddle-point optimization problem. Subsequently, a distributed primal–dual hybrid gradient algorithm (dPDHG) for quantile regression is proposed, and its convergence is analyzed.

3.1 Diffusion-based distributed estimation framework

Since the main task of quantile regression is to estimate the parameter vector $\boldsymbol{w}_0$, we introduce a diffusion-based distributed estimation framework. In this framework, each node solves its local optimization problem using the subsequently proposed algorithm. The nodes then share their intermediate results with neighboring nodes to collaboratively solve the global quantile regression problem.

By definition in Equation 4, the local cost function of node k can be further expressed as a combination of the local cost functions of its neighboring nodes, i.e., $J_k^{\text{loc}}(\bar{\boldsymbol{\omega}}) = \sum_{l \in \mathcal{N}_k} c_{l,k} \tilde{J}_l^{\text{loc}}(\bar{\omega})$, with $\tilde{J}_l^{\text{loc}}(\bar{\omega}) = \sum_{i=1}^{M} \rho_\tau(s_{i,l} - \boldsymbol{a}_{i,l}^\top \bar{\boldsymbol{\omega}}) + \lambda M \|\boldsymbol{w}\|_1$. Therefore, each node obtains its estimate $\bar{\boldsymbol{\omega}}_k$ by combining its newly generated estimate $\boldsymbol{\varphi}_k$ with the estimates received from its neighboring nodes:

$$\bar{\boldsymbol{\omega}}_k = \sum_{l \in \mathcal{N}_k} c_{lk} \boldsymbol{x}_l \in \mathbb{R}^{L+1} \text{ with } \boldsymbol{x}_l = \operatorname{argmin}_{\boldsymbol{x}} \tilde{J}_l^{\text{loc}}(\boldsymbol{x}). \tag{6}$$

Based on this idea, we will use a distributed strategy in the following subsections to solve the problem (Equation 3).

This study focuses on a diffusion-based framework for decentralized quantile regression. The integration of consensus-based strategies with primal–dual hybrid gradient methods for distributed quantile regression, which presents significant algorithmic challenges, is left for future investigation.

3.2 Dual problem and saddle point optimization

By definition, we split $\tilde{J}_l^{loc}(\boldsymbol{x})$ into $\tilde{J}_l^{loc}(\boldsymbol{x}) = g_l(\boldsymbol{x}) + \lambda M f(\boldsymbol{x})$ with $g_l(\boldsymbol{x}) = \sum_{i=1}^{M} \rho_\tau(s_{i,l} - \boldsymbol{a}_{i,l}^\top \boldsymbol{x})$ and $f(\boldsymbol{x}) = \sum_{i=1}^{L} |x_i|$. For the local optimization problem $\min_{\boldsymbol{x}} \tilde{J}_l^{\text{loc}}(\boldsymbol{x})$, the check function $\rho_\tau(v)$ is not differentiable at the origin. To address this, we adopt its conjugate function $\rho_\tau^\star(v)$, which allows us to express the problem equivalently. The conjugate function is defined as

$$\rho_\tau^\star(v) = \sup_u \{uv - \rho_\tau(u)\} = \begin{cases} 0 & \text{if } \tau - 1 \le v < \tau, \\ \infty & \text{otherwise.} \end{cases}$$

Using the conjugate function $\rho_\tau^\star(v)$, we can express $\rho_\tau(v)$ as

$$\rho_\tau(v) = \sup_u \{uv - \rho_\tau^\star(u)\} = \sup_{\tau - 1 \le u < \tau} uv.$$

This suggests that $g(\boldsymbol{x})$ can be expressed as

$$g_l(\boldsymbol{x}) = \max_{\boldsymbol{y}_l} \langle \boldsymbol{y}_l, \boldsymbol{s}_l - \boldsymbol{A}_l \boldsymbol{x} \rangle - \mathcal{I}_\tau(\boldsymbol{y}_l), \tag{7}$$

where $\boldsymbol{A}_l = [\boldsymbol{a}_{1,l}, \ldots, \boldsymbol{a}_{M,l}]^\top$, $\boldsymbol{s}_l = [s_{1,l}, \ldots, s_{M,l}]^\top$, $\boldsymbol{y}_l = [y_{1,l}, \ldots, y_{M,l}]^\top$, and $\mathcal{I}_\tau(\boldsymbol{y})$ is an indicator function given by

$$\mathcal{I}_\tau(\boldsymbol{y}) = \begin{cases} 0, & \tau - 1 \le y_i < \tau, \ \forall i \\ \infty, & \text{else.} \end{cases}$$

We can obtain a dual problem associated with the original function minimization problem $\min_{\boldsymbol{x}} \tilde{J}_l^{\text{loc}}(\boldsymbol{x})$, which can be expressed as follows:

$$\min_{\boldsymbol{x}_k} \max_{\boldsymbol{y}_k} \{ \langle \boldsymbol{y}_k, \boldsymbol{s}_k - \boldsymbol{A}_k \boldsymbol{x}_k \rangle - \mathcal{I}_\tau(\boldsymbol{y}_k) + M\lambda f(\boldsymbol{x}_k) \}. \tag{8}$$

This indicates that the global optimization problem (Equation 3) can be formulated as

$$\min_{\{\boldsymbol{x}_k\}} \max_{\{\boldsymbol{y}_k\}} \sum_{k=1}^{K} \sum_{l \in \mathcal{N}_k} c_{l,k} \left(\langle \boldsymbol{y}_l, \boldsymbol{s}_l - \boldsymbol{A}_l \boldsymbol{x}_l \rangle - \mathcal{I}_\tau(\boldsymbol{y}_l) + KM\lambda f(\boldsymbol{x}_l) \right), \tag{9}$$

which has a standard saddle-point optimization problem expression with the primal variable $\boldsymbol{x}_k$ and the dual variable $\boldsymbol{y}_k$, where

TABLE 1 dPDHG.

Input: $\{\tau, \lambda, s_{k,i}, \boldsymbol{b}_{k,i}, \quad i = 1, \ldots, M, \quad k = 1, \ldots, K\}$
1: **Initialize:** $\mu, \eta, \boldsymbol{x}_k(0), \boldsymbol{y}_k(0)$
2: **for** $n = 1, 2, \ldots, T$ **do**
3: **for** $k = 1, 2, \ldots, K$ **do**
4: $\boldsymbol{x}_k(n+1) = \text{prox}_{\mu KM\lambda f}(\boldsymbol{x}_k(n) + \mu(\boldsymbol{A}_k^\top \boldsymbol{y}_k(n))$
5: $\bar{\boldsymbol{x}}_k(n+1) = 2\boldsymbol{x}_k(n+1) - \boldsymbol{x}_k(n)$
6: **end for**
7: **for** $k = 1, 2, \ldots, K$ **do**
8: $\boldsymbol{\omega}_k(n+1) = \sum_{l \in \mathcal{N}_k} c_{lk} \bar{\boldsymbol{x}}_l(n+1)$
9: $\boldsymbol{y}_k(n+1) = \text{prox}_{\eta \mathcal{I}_\tau}(\boldsymbol{y}_k(n) - \eta(\boldsymbol{A}_k \boldsymbol{\omega}_k(n+1) - \boldsymbol{s}_k))$
10: **end for**
11: **end for**

$k = 1, \ldots, K$. Note that we are primarily concerned with the first L elements of the vector $\boldsymbol{x}$, which can be interpreted as the estimate of $\boldsymbol{w}_0$. Moreover, $f(\boldsymbol{x})$ is defined as $f(\boldsymbol{x}) = \sum_{i=1}^{L} |x_i|$.

3.3 Algorithmic principles and derivations of the dPDHG

This section presents the algorithmic principles and derivations of the proposed dPDHG to solve Equation 9. Its basic framework includes the following steps (a)–(d), which involve iterative updates of the primal and dual variables $\{\boldsymbol{x}_k(n), \boldsymbol{y}_k(n), \quad k = 1, \ldots, K, \quad n = 1, \ldots\}$:

(a) Update the primal variable $\boldsymbol{x}_k(n)$, $k = 0, 1, \ldots, K$:

$$\boldsymbol{x}_k(n+1) = \text{prox}_{\mu KM\lambda f}\left(\boldsymbol{x}_k(n) + \mu\left(\boldsymbol{A}_k^\top \boldsymbol{y}_k(n)\right)\right),$$

where $\text{prox}_{\mu KM\lambda f}(\boldsymbol{z})$: $\mathbb{R}^{L+1} \to \mathbb{R}^{L+1}$ is an element-wise proximal operator defined by $prox_f(v) = \text{argmin}_x(f(x) + (1/2)\|x - v\|_2^2)$. It is readily deduced that

$$\left[\text{prox}_{\mu KM\lambda f}(\boldsymbol{z})\right]_i = \begin{cases} \text{sign}(z_i)\max\left(|z_i| - \mu KM\lambda, 0\right), & \\ \qquad\qquad\qquad\qquad i = 1, \ldots, L & \\ z_i, \quad i = L+1 & \end{cases},$$

with z_i being the i-th element of $\boldsymbol{z}$.

(b) Update the auxiliary variable $\bar{\boldsymbol{x}}_k(n)$, $k = 1, \ldots, K$:

$$\bar{\boldsymbol{x}}_k(n+1) = 2\boldsymbol{x}_k(n+1) - \boldsymbol{x}_k(n).$$

(c) Each node k combines its newly generated estimate $\bar{\boldsymbol{x}}_k(n+1)$ with the estimates received from its neighboring nodes, $\bar{\boldsymbol{x}}_l(n+1), l \in \mathcal{N}_k$: $\boldsymbol{\omega}_k(n+1) = \sum_{l \in \mathcal{N}_k} c_{lk} \bar{\boldsymbol{x}}_l(n+1)$.

(d) Update the dual variable $\boldsymbol{y}_k(n)$, $k = 1, \ldots, K$:

$$\boldsymbol{y}_k(n+1) = \text{prox}_{\eta \mathcal{I}_\tau}\left(\boldsymbol{y}_k(n) - \eta\left(\boldsymbol{A}_k \boldsymbol{\omega}_k(n+1) - \boldsymbol{s}_k\right)\right),$$

where one can deduce that the proximal operator $\text{prox}_{\eta \mathcal{I}}(\boldsymbol{z})$ projects $\boldsymbol{z}$ onto the interval $[\tau - 1, \tau]$ element-wise: $[\text{prox}_{\eta \mathcal{I}}(\boldsymbol{z})]_i = \min(\max(z_i, \tau - 1), \tau)$.

(e) Assign global $\boldsymbol{\omega}$ to local $\boldsymbol{x}$ for the next iteration.

The algorithm continues iterating until the stopping criterion is met, typically based on the difference between successive iterates or the duality gap. Finally, we summarize the proposed dPDHG algorithm in Table 1.

3.4 Selection of the primal–dual step sizes

In the above steps, $\{\mu, \eta\}$ are the primal–dual step sizes chosen to ensure the convergence of the dPDHG algorithm. For the standard form of a convex–concave saddle-point optimization problem $\min_{\mathcal{X}} \max_{\mathcal{Y}} \{\langle \mathcal{Y}, \mathcal{A}\mathcal{X} \rangle - g(\mathcal{Y}) + f(\mathcal{X})\}$, the step sizes $\{\mu, \eta\}$ chosen to ensure convergence of the PDHG algorithm are required $\mu\eta\Omega^2 < 1$ (Esser et al., 2010), where $\Omega := \max_{\mathcal{Z} \in \mathbb{R}^L \setminus \{\mathbf{0}\}} \frac{\|\mathcal{A}\mathcal{Z}\|_2}{\|\mathcal{Z}\|_2} = \rho(\mathcal{A}^\top \mathcal{A})$. For the problem (Equation 9) mentioned in this paper, $\mathcal{A}, \mathcal{X}, \mathcal{Y}$ can be treated as $\mathcal{A} = [\boldsymbol{s}_l, -\boldsymbol{A}_l]$, $\mathcal{X} = [1, \boldsymbol{x}_l^\top]^\top$, and $\mathcal{Y} = \boldsymbol{y}_l$, respectively.

Note that, for each n, choosing a small μ and large η results in small dual residual $\boldsymbol{d}_k(n)$ but large primal residual $\boldsymbol{p}_k(n)$, and *vice versa*, where

$$\begin{aligned} \boldsymbol{p}_k(n) &= \boldsymbol{\mathcal{A}}_k^\top \triangle \boldsymbol{y}_k(n) - \frac{1}{\mu} \triangle \boldsymbol{x}_k(n) \\ \boldsymbol{d}_k(n) &= \boldsymbol{\mathcal{A}}_k \triangle \boldsymbol{x}_k(n) - \frac{1}{\eta} \triangle \boldsymbol{y}_k(n) \end{aligned}, \tag{10}$$

where $\triangle \boldsymbol{x}_k(n) = \boldsymbol{x}_k(n) - \boldsymbol{x}_k(n-1)$ and $\triangle \boldsymbol{y}_k(n) = \boldsymbol{y}_k(n) - \boldsymbol{y}_k(n-1)$. Therefore, adaptive strategies—such as those proposed in Goldstein et al. (2015); Chambolle et al. (2024)—can also be employed to balance the progress between the primal and dual updates, thereby enhancing the convergence. If the primal residual $\boldsymbol{p}_k(n)$ is sufficiently large compared to the dual residual $\boldsymbol{d}_k(n)$, for example $\|\boldsymbol{p}_k(n)\|_2 \geq 2\|\boldsymbol{d}_k(n)\|_2$, we increase the primal step size μ by a factor of $(1-\alpha)^{-1}$ and decrease the dual stepsize η by a factor of $1 - \alpha$. If the primal residual is somewhat smaller than the dual residual, we do the opposite. If both residuals are comparable in size, then let the step sizes remain the same on the next iteration. Moreover, when we modify the step size as the iteration continues, we also shrink the adaptivity level to $\alpha \leftarrow \zeta\alpha$, for $\zeta \in (0, 1)$.

4 Simulation examples

We consider a connected network with $K = 30$ nodes that are positioned randomly on a unit square area, with a maximum communication distance of 0.4 unit length. The three non-zero components of the sparse vector $\boldsymbol{w}_0$ having size $L = 18$ are set as 1.0 with their positions randomly selected, while the others are zeros. The weighting matrix $\boldsymbol{C}$ in Equation 5 is chosen according to the metropolis criterion (Cattivelli and Sayed, 2010; Tu and Sayed, 2011), that is,

$$c_{l,k} = \begin{cases} 1/\max\{\deg_l, \deg_k\}, & \text{if } l \in \mathcal{N}_k, l \neq k \\ 1 - \sum_{l \neq k} c_{l,k}, & \text{if } l = k \\ 0, & \text{otherwise} \end{cases},$$

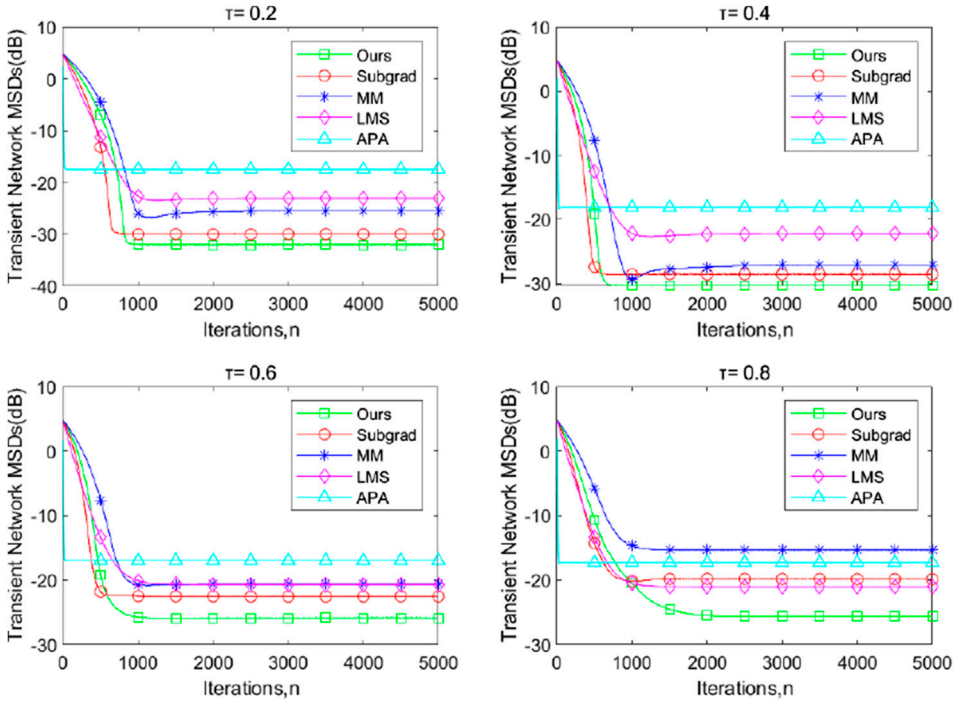

FIGURE 1
The transient network MSDs of the proposed dPDHG algorithm compared with three benchmark distributed algorithms: the Subgrad, the MM, and the LMS, and the APA algorithms for estimating ω_0, where $\epsilon_{k,i}$ is the beta-distributed noise.

where $\deg_k$ denotes the degree of node k (the cardinality of its closed neighborhood).

We begin by evaluating the performance of the distribution estimation for $\boldsymbol{w}_0$. The regressors, $\boldsymbol{b}_{k,i}$, $i = 1, \ldots, M$, and $k = 1, \ldots, K$, are modeled as independent, zero-mean Gaussian random variables in both time and space. Their covariance matrices are assumed to be identity matrices. Moreover, we consider three types of noise $\epsilon_{k,i}$: one following the beta distribution and two following heavy-tailed distributions. Specifically, the heavy-tailed noises are generated from Student's t-distribution with 2 degrees of freedom (dof) and the Cauchy distribution. Beta-distributed noise is produced using the MATLAB command $\texttt{betarnd}(\alpha, \beta) \times 2 - 1$, which maps the standard beta-distributed values from [0,1] to $[-1, 1]$. The Student's t-distributed noise is generated using $\texttt{trnd}(\text{dof}, 1, 1)$, and the Cauchy-distributed noise is generated via the transformation: $\epsilon_{k,i} = \tan((\texttt{rand}(1) - 0.5)\pi)$, where, for each time step i and each node k, a uniform random number in [0,1] is first drawn using $\texttt{rand}(1)$, shifted to the interval $[-0.5, 0.5]$, and then transformed using $\tan(\pi\xi)$ to yield a Cauchy-distributed sample.

Figures 1–3 illustrate the transient-network mean-square deviation (MSD) performance of our proposed dPDHG algorithm, compared with three benchmark algorithms: the subgradient-based algorithm (Subgrad) (Wang and Li, 2018), the majorization–minimization algorithm (MM) (Kai et al., 2023), accelerated proximal-based gradient methods(APG) (Chen and Ozdaglar, 2012), and the least mean squares (LMS) algorithm (Liu et al., 2012). The transient network MSD is defined as $\sum_{k=1}^{K} \mathbb{E}\|\boldsymbol{x}_k(n) - \boldsymbol{w}_0\|^2/K$, and the results are presented for different quantile levels, $\tau = \{0.2, 0.4, 0.6, 0.8\}$; Figures 1–3 are presented under the presence of beta-distributed noise, t-distributed noise, and Cauchy noise, respectively.

Across all quantile levels, our algorithm demonstrates superior convergence properties, achieving the lowest MSD values at steady state, compared to the other algorithms. The APG and LMS algorithms exhibit the highest MSD, indicating poor adaptation to the beta-distributed noise, t-distributed noise, and the Cauchy noise. The MM algorithm outperforms Subgrad but converges to higher MSD values than our approach. Subgrad shows moderate performance but struggles to maintain consistent improvements across iterations. These results highlight the robustness and efficiency of the proposed DQR algorithm in addressing distributed quantile regression tasks.

We further consider a practical application in spectrum estimation for a narrow-band source. A peaky spectrum can be modeled by an L-order sparse AR process (Liu et al., 2012; Schizas et al., 2009): $\theta_i = -\sum_{l=1}^{L} \pi_l \theta_{i-l} + \varepsilon_i$, where ε_i is a noise and $\{\pi_1, \ldots, \pi_L\}$ are the AR coefficients. The source propagates to sensor k via a transmission channel modeled by a $\bar{L}_k$-order FIR filter, yielding an observation $x_{k,i} = \sum_{l=0}^{\bar{L}_k - 1} \varsigma_{k,l} \theta_{i-l} + \epsilon_{k,i}$, where $\epsilon_{k,i}$ is an additive sensing noise and $\{\varsigma_{k,l}\}$ are the FIR coefficients. It is readily deduced that $x_{k,i}$ can be rewritten as an autoregressive moving average (ARMA) process (see Appendix A):

$$x_{k,i} = -\sum_{l=1}^{L} \pi_l x_{k,i-l} + \sum_{j=1}^{L+\bar{L}_k+1} \zeta_j \eta_{k,i-j}, \tag{11}$$

where the MA coefficients $\{\zeta_j\}$ and the variance of the white noise $\eta_{k,i}$ depend on $\varsigma_{k,l}, \pi_l$ and the variances of the noise terms ε_i and $\epsilon_{k,i}$.

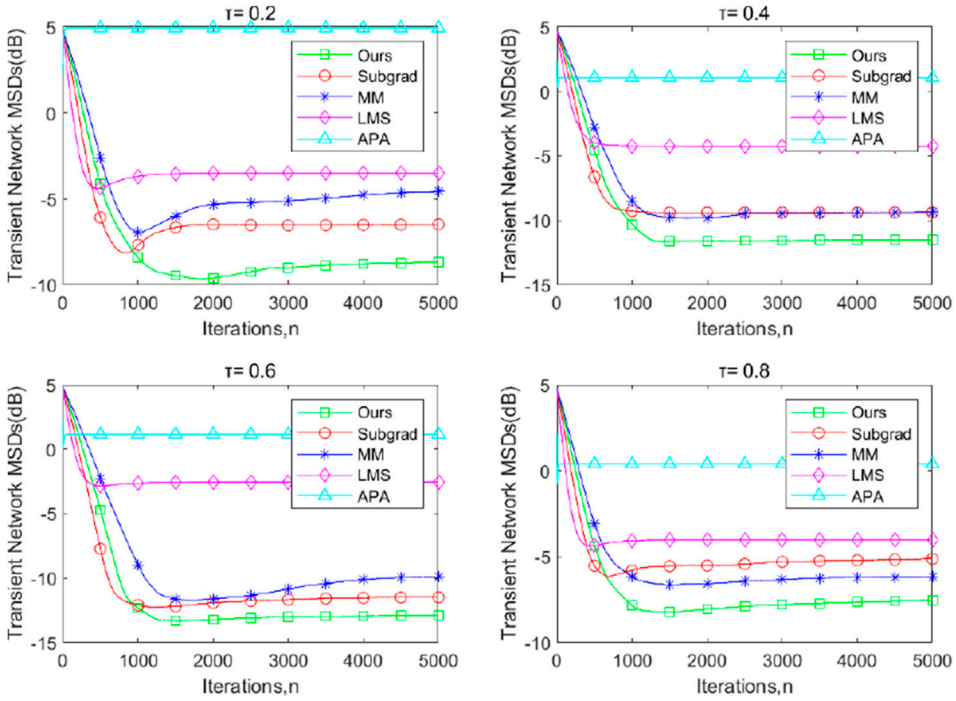

FIGURE 2
The transient-network MSDs of the proposed dPDHG algorithm compared with three benchmark distributed algorithms: the Subgrad, the MM, and the LMS, and the APA algorithms for estimating ω_0, where $\epsilon_{k,i}$ is the t-distribution noise.

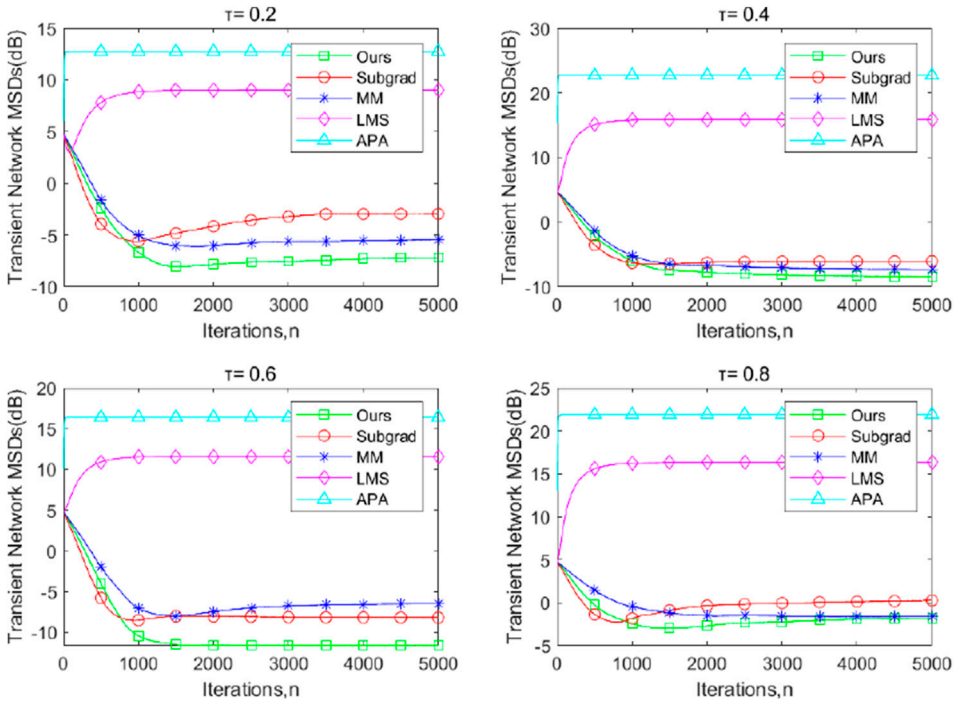

FIGURE 3
The transient network MSDs of the proposed dPDHG algorithm compared with three benchmark distributed algorithms: the Subgrad, the MM, and the LMS, and the APA algorithms for estimating ω_0, where $\epsilon_{k,i}$ is the Cauchy noise.

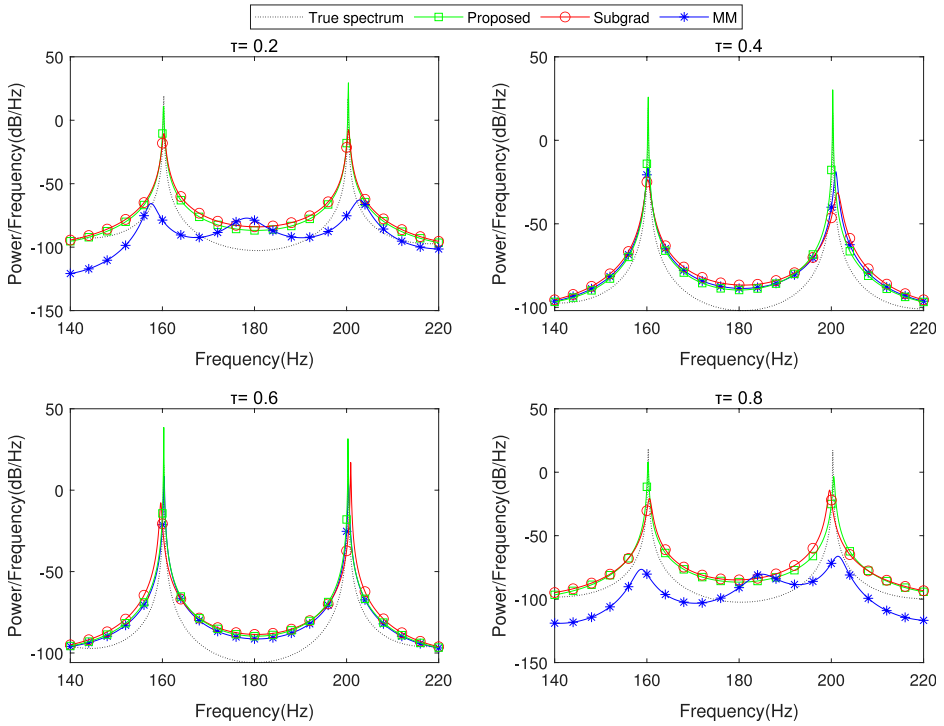

FIGURE 4
The spectrum estimation results of the proposed dPDHG algorithm compared with two benchmark distributed algorithms: Subgrad and MM. The comparison is conducted in the presence of beta-distributed noise $\epsilon_{k,i}$.

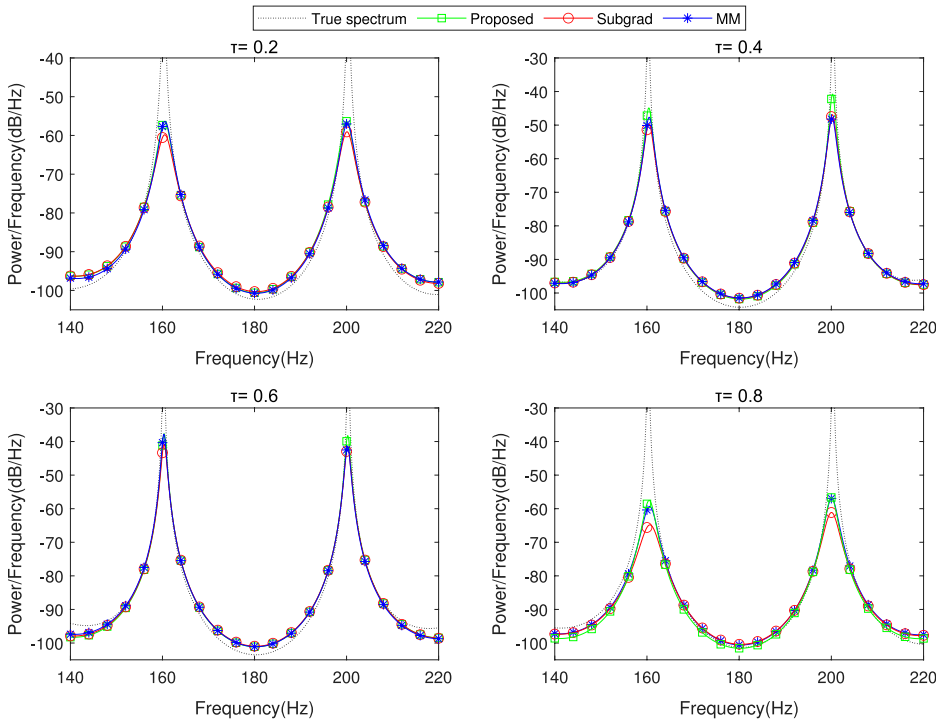

FIGURE 5
The spectrum estimation results of the proposed dPDHG algorithm compared with two benchmark distributed algorithms: Subgrad and MM. The comparison is conducted in the presence of t-distributed noise $\epsilon_{k,i}$.

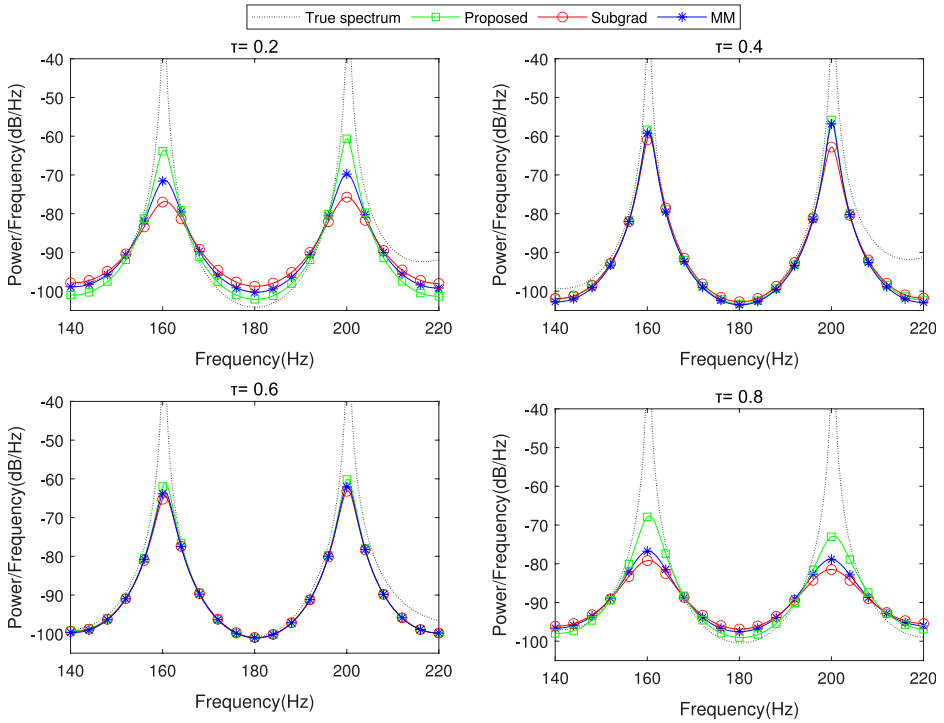

FIGURE 6
The spectrum estimation results of the proposed dPDHG algorithm compared with two benchmark distributed algorithms: Subgrad and MM. The comparison is conducted in the presence of Cauchy noise $\epsilon_{k,i}$.

For more details, refer to Appendix A. To determine the spectral contents of the source, the MA term in Equation 11 can be treated as an observation noise, and then the spectral peaks of the source can be obtained by estimating the AR coefficients π_l. By letting $\boldsymbol{b}_{k,i} = [-x_k(i-1), \ldots, -x_k(i-L)]^\top$, $s_{k,i} = x_{k,i}$ and $\boldsymbol{w}_0 = [\pi_1, \ldots, \pi_L]$, the problem of spectrum estimation fits our model (Equation 2) and becomes that of the distributed estimation as mentioned above.

Figures 4–6 compare the true source spectrum with the estimated results averaged across K nodes under three distinct channel noise distributions: beta-distributed, t-distributed, and Cauchy noise ($\epsilon_{k,i}$). The noise sequences are generated using the methodology outlined previously, maintaining consistent simulation parameters. In this simulation, we configure the AR coefficients and channel parameters as follows:

- The peaked spectrum is generated from a 20th-order autoregressive (AR) model (order $L = 20$). The true spectral peaks are located at normalized frequencies corresponding to 160 Hz and 200 Hz.
- The multipath channels have a fixed length of $\bar{L}_k = 2$ for all $k = 1, \ldots, K$.
- The FIR channel coefficients $\{\varsigma_{k,1}, \varsigma_{k,2}\}$ are generated using the `randn(2,1)` command in MATLAB, producing standard normal random values.
- The AR process noise ε_i is modeled as zero-mean Gaussian random variables with variance 10^{-4}.

As shown in these figures, our algorithm closely matches the true spectrum across all tested quantile levels, achieving higher estimation accuracy than the benchmark algorithms. These results highlight the robustness and precision of the proposed dPDHG algorithm in estimating the spectrum of narrow-band sources under challenging noise conditions.

5 Conclusion

This paper investigated distributed robust estimation in sensor networks and introduced a distributed quantile regression algorithm based on the primal–dual hybrid gradient method. The proposed algorithm effectively addresses the challenge of non-differentiability in the optimization problem by iteratively identifying the saddle point of a convex–concave objective.

Additionally, it mitigates the issue of slow convergence commonly associated with such problems. The method demonstrates robustness, scalability, and suitability for processing large-scale data distributed across sensor networks.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

ZQ: Writing – review and editing. ZL: Writing – original draft.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Publisher's note

References

Bazzi, A., and Chafii, M. (2023). On integrated sensing and communication waveforms with tunable papr. *IEEE Trans. Wirel. Commun.* 22, 7345–7360. doi:10.1109/TWC.2023.3250263

Bazzi, A., Slock, D. T. M., and Meilhac, L. (2017). "A Newton-type forward backward greedy method for multi-snapshot compressed sensing," in *2017 51st asilomar conference on signals, systems, and computers*, 1178–1182. doi:10.1109/ACSSC.2017.8335537

Ben Taieb, S., Huser, R., Hyndman, R. J., and Genton, M. G. (2016). Forecasting uncertainty in electricity smart meter data by boosting additive quantile regression. *IEEE Trans. Smart Grid* 7, 2448–2455. doi:10.1109/TSG.2016.2527820

Cade, B. S., and Noon, B. R. (2003). A gentle introduction to quantile regression for ecologists. *Front. Ecol. Environ.* 1, 412–420. doi:10.1890/1540-9295(2003)001[0412:agitqr]2.0.co;2

Cattivelli, F. S., and Sayed, A. H. (2010). Diffusion LMS strategies for distributed estimation. *IEEE Trans. Signal Process.* 58, 1035–1048. doi:10.1109/tsp.2009.2033729

Chambolle, A., Delplancke, C., Ehrhardt, M. J., Schönlieb, C.-B., and Tang, J. (2024). Stochastic primal–dual hybrid gradient algorithm with adaptive step sizes. *J. Math. Imaging Vis.* 66, 294–313. doi:10.1007/s10851-024-01174-1

Chen, A. I., and Ozdaglar, A. (2012). "A fast distributed proximal-gradient method," in *2012 50th annual allerton Conference on communication, control, and computing (allerton)* (IEEE).

Cheng, Y., and Kuk, A. Y. C. (2024). MM algorithms for statistical estimation in quantile regression

Delamou, M., Bazzi, A., Chafii, M., and Amhoud, E. M. (2023). "Deep learning-based estimation for multitarget radar detection," in *2023 IEEE 97th vehicular technology conference (VTC2023-Spring)*, 1–5. doi:10.1109/VTC2023-Spring57618.2023.10200157

Esser, E., Zhang, X., and Chan, T. F. (2010). A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J. Imaging Sci.* 3, 1015–1046. doi:10.1137/09076934x

Gao, M., Niu, Y., and Sheng, L. (2022). Distributed fault-tolerant state estimation for a class of nonlinear systems over sensor networks with sensor faults and random link failures. *IEEE Syst. J.* 16, 6328–6337. doi:10.1109/jsyst.2022.3142183

Goldstein, T., Li, M., and Yuan, X. (2015). "Adaptive primal-dual splitting methods for statistical learning and image processing," in *Proceedings of the 29th international conference on neural information processing systems - volume 2* (Cambridge, MA, USA: MIT Press), 2089–2097.

Hovine, C., and Bertrand, A. (2024). A distributed adaptive algorithm for non-smooth spatial filtering problems in wireless sensor networks. *IEEE Trans. Signal Process.* 72, 4682–4697. doi:10.1109/TSP.2024.3474168

Hüttel, F. B., Peled, I., Rodrigues, F., and Pereira, F. C. (2022). Modeling censored mobility demand through censored quantile regression neural networks. *IEEE Trans. Intelligent Transp. Syst.* 23, 21753–21765. doi:10.1109/TITS.2022.3190194

Kai, B., Huang, M., Yao, W., and Dong, Y. (2023). Nonparametric and semiparametric quantile regression via a New MM algorithm. *J. Comput. Graph. Stat.* 32, 1613–1623. doi:10.1080/10618600.2023.2184374

Lee, J., Tepedelenlioglu, C., and Spanias, A. (2018). Consensus-based distributed quantile estimation in sensor networks. *arXiv Prepr. arXiv:1805.00154*.

Lee, J., Tepedelenlioglu, C., Spanias, A., and Muniraju, G. (2020). ngermanDistributed quantiles estimation of sensor network measurements. *Int. J. Smart Secur. Technol.* 7, 38–61. doi:10.4018/ijsst.2020070103

Liu, Y., Li, C., and Zhang, Z. (2012). Diffusion sparse least-mean squares over networks. *IEEE Trans. Signal Process.* 60, 4480–4485. doi:10.1109/tsp.2012.2198468

Mirzaeifard, R., Venkategowda, N. K. D., Gogineni, V. C., and Werner, S. (2024). Smoothing admm for sparse-penalized quantile regression with non-convex penalties. *IEEE Open J. Signal Process.* 5, 213–228. doi:10.1109/OJSP.2023.3344395

Njima, W., Bazzi, A., and Chafii, M. (2022). Dnn-based indoor localization under limited dataset using gans and semi-supervised learning. *IEEE Access* 10, 69896–69909. doi:10.1109/ACCESS.2022.3187837

Patidar, V. K., Wadhvani, R., Shukla, S., Gupta, M., and Gyanchandani, M. (2023). "Quantile regression comprehensive in machine learning: a review," in *2023 IEEE international students' conference on electrical, electronics and computer science (SCEECS)*, 1–6. doi:10.1109/SCEECS57921.2023.10063026

Schizas, I. D., Mateos, G., and Giannakis, G. B. (2009). Distributed LMS for consensus-based in-network adaptive processing. *IEEE Trans. Signal Process.* 57, 2365–2382. doi:10.1109/tsp.2009.2016226

Swain, R., Khilar, P. M., and Dash, T. (2018). Fault diagnosis and its prediction in wireless sensor networks using regressional learning to achieve fault tolerance. *Int. J. Commun. Syst.* 31. doi:10.1002/dac.3769

Tu, S.-Y., and Sayed, A. H. (2011). Mobile adaptive networks. *IEEE J. Sel. Top. Signal Process.* 5, 649–664. doi:10.1109/jstsp.2011.2125943

Waldmann, E. (2018). Quantile regression: a short story on how and why. *Stat. Model.* 18, 203–218. doi:10.1177/1471082x18759142

Wan, C., Lin, J., Wang, J., Song, Y., and Dong, Z. Y. (2017). Direct quantile regression for nonparametric probabilistic forecasting of wind power generation. *IEEE Trans. Power Syst.* 32, 2767–2778. doi:10.1109/TPWRS.2016.2625101

Wang, H., and Li, C. (2018). Distributed quantile regression over sensor networks. *IEEE Trans. Signal Inf. Process. over Netw.* 4, 338–348. doi:10.1109/TSIPN.2017.2699923

Wang, Y., and Lian, H. (2023). On linear convergence of ADMM for decentralized quantile regression. *IEEE Trans. Signal Process.* 71, 3945–3955. doi:10.1109/TSP.2023.3325622

Appendix

Appendix A spectrum estimation

Since θ_i is generated by an AR process, we substitute the AR model into the FIR filter equation. For each θ_{i-l}, replace it with its corresponding AR expression: $\theta_{i-l} = -\sum_{l'=1}^{L} \pi_{l'} \theta_{i-l-l'} + \varepsilon_{i-l}$. Thus, the observation model $x_{k,i} = \sum_{l=0}^{\bar{L}_k-1} \varsigma_{k,l} \theta_{i-l} + \epsilon_{k,i}$ becomes

$$\begin{aligned} x_{k,i} &= \sum_{l=0}^{\bar{L}_k-1} \varsigma_{k,l} \left(-\sum_{l'=1}^{L} \pi_{l'} \theta_{i-l-l'} + \varepsilon_{i-l} \right) + \epsilon_{k,i} \\ &= -\sum_{l=1}^{L} \pi_l x_{k,i-l} + \sum_{j=1}^{L+\bar{L}_k+1} \zeta_j \eta_{k,i-j}, \end{aligned} \tag{A1}$$

where $\zeta_j = [\boldsymbol{\zeta}]_j$ and $\eta_{k,i-j} = [\boldsymbol{\eta}_{k,i}]_j$ are defined as follows:

$$\boldsymbol{\zeta} \triangleq \left[1, \pi_1, \ldots, \pi_L, \varsigma_{k,0}, \varsigma_{k,1}, \ldots, \varsigma_{k,\bar{L}_k-1}\right]^\top,$$

$$\boldsymbol{\eta}_{k,i} \triangleq \left[\epsilon_{k,i}, \epsilon_{k,i-1}, \ldots, \epsilon_{k,i-L}, \varepsilon_i, \varepsilon_{i-1}, \ldots, \varepsilon_{i-\bar{L}_k+1}\right]^\top,$$

both vectors having a length of $L + \bar{L}_k + 1$.

Appendix Equation A.1 expresses $x_k(t)$ as a combination of past observations (the AR part) and past noise (the MA part), thus forming the desired ARMA process, as shown in Equation 11.